Compound Movement Recognition using Dynamic Movement Primitives *

Ali H. Kordia[1,2] and Francisco S. Melo[1,2]

[1] INESC-ID, Lisbon, Portugal
[2] Instituto Superior Técnico, University of Lisbon, Portugal
ali.kordia@tecnico.ulisboa.pt,fmelo@inesc-id.pt

Abstract. This paper proposes a method for recognizing compound trajectories. Given a library of pre-learned movements described using dynamic movement primitives, our approach is able to break down an observed trajectory into its individual components, each of which is a segment of one of the movements in the library. We build on previous work that uses critical points for movement recognition and prediction, and extend it to handle trajectories comprising multiple segments from possibly different movements. Our approach assumes that each segment in the observed trajectory is either the initial segment of a new primitive or the continuation of the previous segment. Then, given a partial trajectory, our method is able to predict the most likely next target—i.e., the end-point of the movement currently being executed, if the latter is executed to the end. By using an effective search tree, our approach is able to run at execution time and provide an efficient way for action recognition and prediction, which has applications in human-robot interaction scenarios. We validate our approach both in simulation and in a human-robot interaction scenario involving the Baxter robot.

Keywords: Movement recognition · Movement prediction · Compound movements · Dynamic movement primitives

1 Introduction

The interaction between humans and robots has become more widespread. Robots act to cooperate with humans in all areas of our daily life. The ability of a robot to understand humans and respond quickly during the interaction is, therefore, a fundamental skill that everyday robotic platforms should have. Towards the development of such skills, robots should recognize and understand the actions of humans, even if they are complex and comprise several different movements.

* This work was partially supported by national funds through FCT, Fundação para a Ciência e a Tecnologia, under projects UIDB/50021/2020 and PTDC/CCI-COM/7203/2020. The first author acknowledges the Ph.D. grant from the Global Platform for Syrian Students. The authors thank Miguel Vasco for useful discussions and revisions of the document.

During infancy, humans build a "library" of new movements by learning to combine simpler movements [5]. This method of learning complex movements allows humans to learn how to carry out complex tasks. The ability to combine simple motions into more complex ones has also been a topic of intense research in the robotics community [6, 8, 14, 15].

In this paper, we look at the converse problem: can we recognize a complex movement as the combination of several simpler movements and, if so, which? The ability to break down a movement into its simpler components is fundamental not only for recognition purposes but also if the robot wants to replicate the observed movement: instead of having to learn the new movement from scratch, it can use simpler movements it already knows (and which were recognized by observation) and compose them to perform a more complex one [6].

There is a significant volume of work on movement recognition and prediction. For example, some works propose the use of *hidden Markov models* (HMM) to recognize motions [1,9,16,19,20] or to compute a probability distribution over a set of pre-learned trajectories [2]. Yoon et al [20] and Kordia and Melo [7] use specific features of the observed trajectory to facilitate recognition. However, all previous works assume that the observed trajectory is atomic and corresponds (or is close) to one of a set of pre-learned trajectories. Such methods are not designed to deal with compound trajectories.

Some works do address complex movement recognition, assuming that the observed movement is obtained as the combination of *complete* simpler motions [3]. Maeda et al [12] use a probabilistic model (I-ProMPs) to distinguish movements, whereby the observed path is approached with the closest human action to it. Ma et al [11] work at the task level: the observed movements of the human body are divided into parts that may overlap. An algorithm is built to capture these parts by extracting them from video frames. A bag-of-words representation is then used to distinguish the partial movements, using the temporal and spatial relationship between the parts.

In this work, we do not assume that the observed motion is composed of full atomic movements. Instead, we accommodate the case where the observed movement arises as the composition of several *movement segments*, possibly coming from different movement primitives. We assume that each such segment is either the initial segment of a new movement primitive or the continuation of the previous segment. We build on an existing method for movement recognition that overcomes movement alignment problems by using *critical points* [7], and discover the simplest representation of the observed movement as the combination of segment from pre-learned movement primitives. We use *dynamic movement primitives* (DMP) [4] as our preferred movement representation.

Our method identifies the observed movement, whether it is new or compound, and predicts its most likely target, should the current movement be executed all the way to the end. The recognition of the observed movement as the combination of simpler movements can then be used in a framework for movement combination and optimization [8, 10, 17, 18].

2 Recognition and prediction using critical points

In this section, we go over our previous recognition framework [7] upon which our approach is built. In our previous work, we used the difference in change ratios of both linear velocity and angle at critical points to match previously learned trajectories with a trajectory being observed. The use of critical points alleviates the need for time aligning the trajectories, while the use of change ratios makes the recognition robust to transformations of the trajectory such as scaling or rotation.

2.1 Motion recognition

A movement in the library is denoted as a trajectory $\mathbf{y}_{0:T} = \{y(t), t = 0, \ldots, T\}$. A *critical point* [13] is any local minimum of the function $h(t)$ defined as[3]

$$h(t) = \|\dot{y}(t)\|^2 + \|\ddot{y}(t)C_t\|^2. \tag{1}$$

Let $\tau_0, \ldots, \tau_{M+1}$ denote the critical points of the observed trajectory. Following Kordia and Melo [7], to match a segment m of the observed trajectory $\mathbf{y}_{0:t}$ with a movement $\mathbf{y}_n$ in the library, we compute the difference in ratios of velocity change between the segment m of $\mathbf{y}_{0:t}$ and the corresponding segment of $\mathbf{y}_n$, i.e.,

$$\epsilon_v^{(m)}(\mathbf{y}_{0:t}, \mathbf{y}_n) = \bar{\delta}_v^{(m)}(\mathbf{y}_n) - \bar{\delta}_v^{(m)}(\mathbf{y}_{0:t}).$$

Similarly, we compute the difference in ratios of angle change in the segments m of $\mathbf{y}_{0:t}$ and $\mathbf{y}_n$ as

$$\epsilon_\theta^{(m)}(\mathbf{y}_{0:t}, \mathbf{y}_n) = \bar{\delta}_\theta^{(m)}(\mathbf{y}_n) - \bar{\delta}_\theta^{(m)}(\mathbf{y}_{0:t}).$$

The local error at the mth critical point with the nth DMP in the library is a function of both differences, and given by

$$\epsilon^{(m)}(\mathbf{y}_{0:t}, \mathbf{y}_n) = \|\epsilon_v^{(m)}(\mathbf{y}_{0:t}, \mathbf{y}_n)\| + \|\epsilon_\theta^{(m)}(\mathbf{y}_{0:t}, \mathbf{y}_n)\|. \tag{2}$$

Finally, the accumulated error for the nth DMP in the library is

$$E(\mathbf{y}_{0:t}, \mathbf{y}_n) = \sum_{m=1}^{M_*} \epsilon^{(m)}(\mathbf{y}_{0:t}, \mathbf{y}_n), \tag{3}$$

where M_* is the total number of critical points observed so far.

The error $E(\mathbf{y}_{0:t}, \mathbf{y}_n)$ is calculated for the complete set of N movements in the library. From such computation, it is possible to build a recognition matrix with the accumulated error as each new critical point is observed. The movement selected (recognized) is the one with smallest accumulated error.

[3] C_t is a time-dependent weighting factor. In our experiments, we used $C_t = 1$.

2.2 Motion prediction

Motion prediction employs the recognition results to predict the endpoint of the observed trajectory. The original paper proposes two approaches to predict the endpoint of an observed trajectory, but in this paper we consider only one of the two, which is also based on extrapolating the trajectory beyond the observed critical points using change ratios—much like those considered for recognition [7].

To predict the end point of a partial movement recognized as being $\mathbf{y}_n$ in the library, we assume that the ratio of change between the current critical point (say m) and the end of the trajectory and between the current critical point and the beginning of the trajectory should be constant in the two executions of the movement—i.e., the one being observed and the one in the library. In other words,

$$\frac{y_n(\tau_{M+1}) - y_n(\tau_m)}{y_n(\tau_m) - y_n(\tau_0)} = \frac{y(\tau_{M+1}) - y(\tau_m)}{y(\tau_m) - y(\tau_0)}. \tag{4}$$

Rearranging the terms, we can compute the predicted endpoint $\hat{y}_n(\tau_{M+1})$ as

$$\hat{y}_n(\tau_{M+1}) = y(\tau_m) + \frac{y(\tau_m) - y(\tau_0)}{y_n(\tau_m) - y_n(\tau_0)} \cdot (y_n(\tau_{M+1}) - y_n(\tau_m)). \tag{5}$$

3 Recognition and prediction of compound movements

We address the problem of movement recognition, when the observed movement is possibly a compound trajectory comprising segments from multiple simpler movements. We henceforth refer to these simpler movements as *primitive movements*. The purpose of our approach is to recognize the individual components forming the observed movement, out of a library of pre-learned (primitive) movements. We assume that each segment in the trajectory is either (i) the initial segment of a possibly different primitive movement; or (ii) the continuation of the previous primitive movement. We consider a segment to be the part of a trajectory lying between two critical points.

Besides recognition, we are also interested in predicting the endpoint of the observed trajectory. Unfortunately, since the trajectory can be formed by an arbitrary number of segments coming from different primitive movements, it is not possible to predict, beforehand, the endpoint of the movement—unless if we know exactly how many and which segments compose the trajectory. As such, we instead predict *the most likely end-point for the current primitive movement*, assuming that it is performed all the way to the end.

Our approach builds on the recognition and prediction framework of Kordia and Melo [7], summarized in Section 2.2, extending such work to consider partial trajectory recognition and prediction. Much like in that work, our method relies on the identification of the the critical points extracted from the observed trajectory, comparing them with the corresponding critical points in the trajectories in the library.

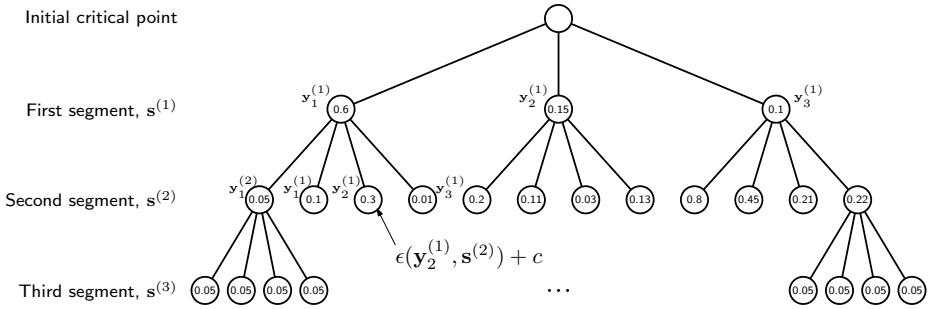


Fig. 1: Example of a search tree. Each node is associated with a segment of a primitive movement in the library, and contains the error associated when comparing the current segment with the primitive movement segment corresponding to that node (the numbers in the nodes).

3.1 Motion recognition

We can interpret the approach of Kordia and Melo [7] as deploying a "sequence search": at each critical point, the current "sequence of segments" is compared with each movement in the library, and recognition thus consists of identifying the movement in the library that best matches the observed "sequence".

In our proposed approach, we instead build a *search tree*. The depth of the tree corresponds to the number of observed segments, and each node corresponds to a primitive movement in the library. The search tree is built as follows:

- Each node of the tree has as many children as there are primitive movements in the library plus one.
- The children corresponding to the primitive movements in the library represent the situation where the current segment is the initial segment of the respective primitive movements.
- The additional child represents the situation where the current segment is a continuation of the current primitive movement.
- Each node in the tree is associated with a segment error, as defined in (2).

Figure 1 shows an example of a possible search tree for the case where the movement library contains only 3 primitive movements. By using the search tree, we can now determine the sequence of primitives that minimizes the total cost. After each new critical point is discovered, the tree grows a level in depth and the node values are computed.

Note that our approach considers the several possible cases: that the current segment corresponds to the beginning of a new primitive movement; or that the current segment is the continuation of the previous primitive movement. Additionally, to discourage the search process from excessively switching between movement primitives, we add a small cost of c to all the nodes corresponding to changes in the movement primitives.

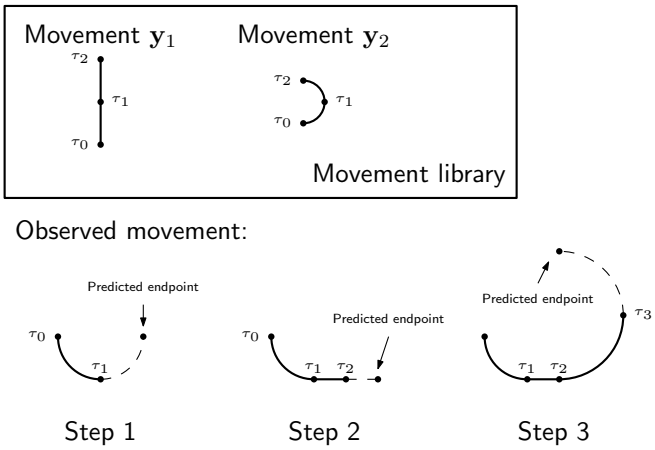


Fig. 2: Example of what the prediction process looks like. After observing a segment and recognizing to which primitive movement such segment "belongs", we predict the endpoint as the endpoint of the corresponding primitive movement.

To render searching the tree a more efficient process, in our approach we do not grow the complete tree simultaneously. Instead, we expand only the nodes corresponding to the path with the smallest average cost per node. While this heuristic for growing the tree still leads to the expansion of some unnecessary nodes, it is significantly more efficient than performing a full breadth-first search.

3.2 Motion prediction

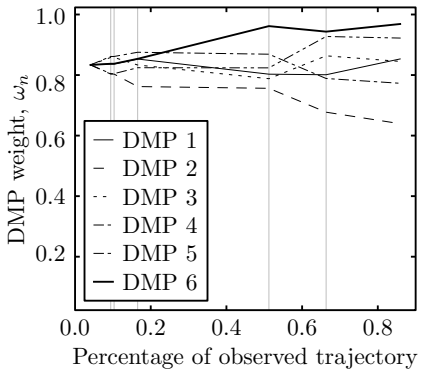
As previously discussed, the observed trajectory can be formed by an arbitrary number of segments coming from different primitive movements. As such, it is not possible to predict, beforehand, the endpoint of the complete movement. In our approach, we predict *the most likely end-point for the current primitive movement*, assuming that it is performed all the way to the end.

The prediction process uses the basic approach described in Section 2.2. At each critical point, after associating the most recent segment with a segment from one of the primitive movements in the library, we compute the endpoint of such primitive movement as the predicted endpoint of the trajectory, using (5). The process is illustrated in Fig. 2.

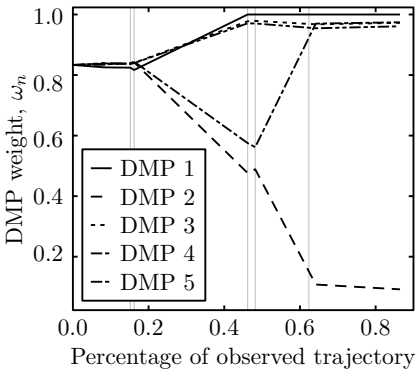
4 Experiments and Results

This section describes the experiments conducted to test the proposed work. We conducted four experiments, each of which is intended to show the effectiveness, accuracy, and features of the proposed approach.

The first experiment illustrates the recognition process during the discovery of new critical points and the accuracy of the prediction of the target point.



(a) Recognition weights for all DMP sequences at successive critical points.



(b) Recognition weights for each DMP at successive critical points when one DMP matches the observed trajectory (from [7]).

Fig. 3: Weights evolution during the recognition process, as larger portions of the trajectory are observed.

The second experiment showcases the ability of our method to recognize the components forming the observed trajectory, and how these are used to predict the endpoint of the trajectory. The third experiment illustrates the scalability of our approach when the library contains a large number of primitive movements. Finally, the fourth experiments showcases the application of our approach in a real world scenario featuring the Baxter robot.

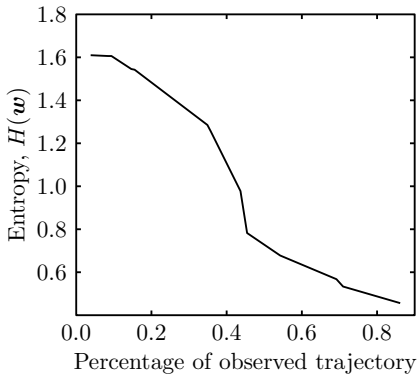
In the first three experiments, we use a 2D simulation environment. The library contains trajectories stored in the form of DMPs. A trajectory is gradually observed, and our approach performs recognition and prediction in runtime.

4.1 Recognition

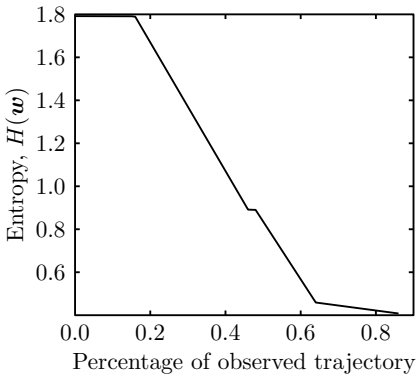
We start by illustrating the ability of our approach to perform recognition when the observed trajectory comprises segments from multiple primitive movements and when it contains segments from a single movement. The results are in Fig. 3.

We notice in Fig. 3a that for some DMPs the recognition weights may first decrease, as the system identifies another primitive as the most likely, and then increase, because the system decides that better recognition is attained by switching to that particular primitive movement. The converse is also observed: the weight of some primitives first increases, as that primitive is the best to explain the observed movement, and then decrease, as switching to another primitive offers better recognition.

To facilitate the analysis of the performance, we depict in Fig. 4 the evolution of the entropy in the recognition weights. We immediately note that the recognition process is slower if the observed trajectory is the composition of more than



(a) Entropy of the prediction with the observation of new critical points.



(b) Entropy of the prediction with the observation of new critical points when one DMP matches the observed trajectory (from [7]).

Fig. 4: Entropy evolution during the recognition process, as larger portions of the trajectory are observed.

one movement, compared with a trajectory comprising a single movement. In fact, comparing the results in Fig. 4a with the results of the recognition using a single trajectory, in Fig. 4b, we note that the entropy decreases more slowly in Fig. 4a. This is due to the fact that, after the currently observed trajectory, we still consider the possibility that the current trajectory may not yet be over and, as such, there may be new primitive movements following the current trajectory. Hence, as we observe new critical points, there is always some level of uncertainty regarding the prediction that is only resolved as the trajectory concludes.

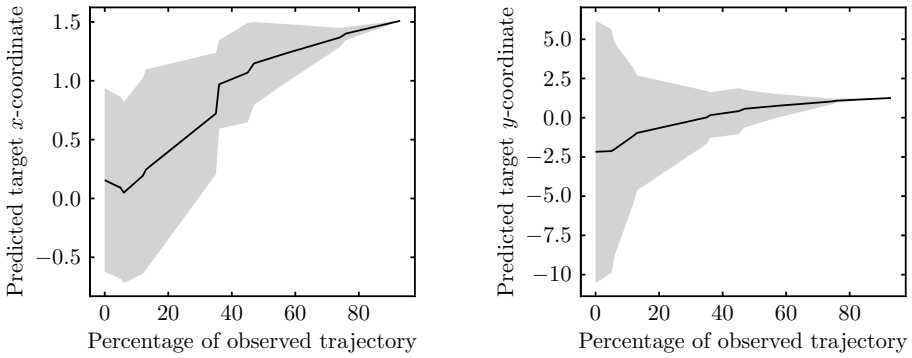
4.2 Prediction

We also tested the ability of our approach to predict the target point of the observed trajectory. We depict in Fig. 5 the evolution of our prediction as more critical points are observed.

Note how the prediction approaches towards the target point, and the uncertainty decreases with the discovery of new critical points. Also, as seen in Fig. 4a, the entropy decreases lowly, which affects the prediction. The prediction process needs to observe the last part of the monitored trajectory to reach the last DMP in the chosen configuration and obtain a correct prediction, so slower convergence is to be expected.

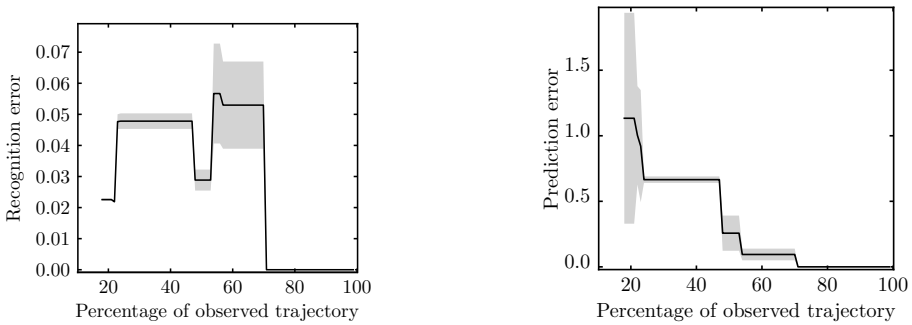
4.3 Accuracy

We now test the ability of our approach to deal with large movement libraries. We compute the recognition error, given by $|\omega_{\max} - \omega_*|$, with $\omega_{\max}$ is the weight



(a) Prediction of the target x coordinate. (b) Prediction of the target y coordinate.

Fig. 5: Uncertainty in the prediction of the target for the two dimensions of the trajectory. The solid line corresponds to the mean predicted target, $\hat{y}(\tau_{M+1})$, while the shaded area corresponds to the variance in the prediction.



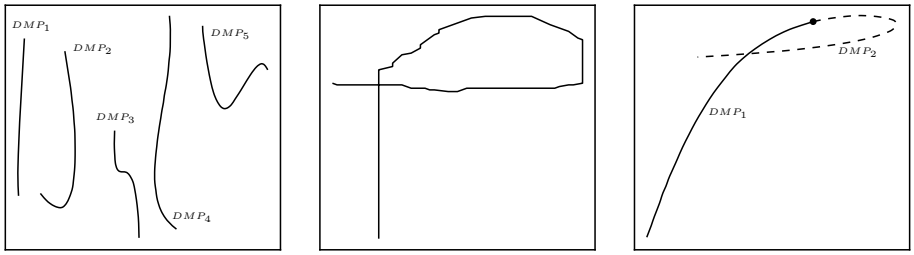
(a) Recognition error. (b) Prediction error.

Fig. 6: The errors in recognition and prediction for the 25 observed trajectories, as a function of the observed trajectory. The solid lines depict the mean error values, while the shaded area depicts the variance in these errors.

associated with the most likely DMP sequence, and ω_* is the weight for the *real* DMP sequence corresponding to the observed trajectory. We also compute the prediction error, given by $\|\hat{y}(\tau_{M+1}) - y(\tau_{M+1})\|$, where $\hat{y}(\tau_{M+1})$ is the predicted endpoint and $y(\tau_{M+1})$ is the actual endpoint.

In Fig. 6 we can see the average prediction and recognition errors averaged over 25 different trajectories, composed from segments of movements from a library of 85 primitive movements.

It is apparent from the figure that, while the prediction error decreases steadily (the trajectory approaches its endpoint, so the prediction errors tend to decrease), the recognition error fluctuates significantly, due to the fact that



(a) DMPs library. (b) Observed trajectory. (c) Recognised DMP.

Fig. 7: Recognizing the "P" letter using two of learned DMPs in the DMPs library, where the trajectory in 7b is constructed using the first DMP and the second one.

when a new segment is observed, the cost for changing the primitive motion generating a segment leads the search algorithm to, sometimes, "delay" the shift to a different primitive.

4.4 Improving the knowledge of the robot

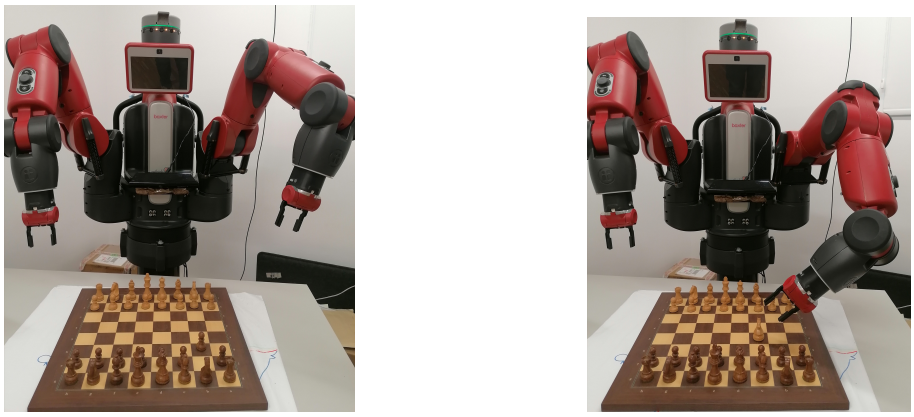
We illustrate a practical application of our approach, where we recognize the letter "P" using a library of number trajectories (or parts thereof). Note, for example, that "P" can be formed by composing the movement for number "1" and part of the movement for number "3" (see Fig. 7). Similarly, the output resulting from observing the trajectory for "B" would be three consecutive partial paths: "1" followed by half of "3" twice.

4.5 Experiment with Robot

In this experiment, Baxter is initially shown the movements of the Pawn (one step forward in the board) and the King (one step sideways in the board) in chess. Baxter then observes a human user performing the movement of the Knight (two steps forward and one step sideways). The Knight's movement can be seen as a composition of two consecutive movements from the Pawn and one of the King, and Baxter successfully recognizes the movement as the composition of the two simpler movements. Baxter could now use the identified sequence of movements to replicate the Knight's movement using, for example, the framework of Kordia and Melo [6].

5 Conclusion

In this work we presented a simple, effective, and robust way to observe a trajectory, predict its goal, and identify the sub-trajectories that compose it. Our method works without training or prior knowledge of the observed trajectory,



(a) (b)

Fig. 8: Illustration of the human-interaction scenario considered in the paper, where BAXTER succeed in anticipation to the user complex motion.

and relies only on the movements in its library. The library serves as an initial knowledge base of the robot, but can be improved as more trajectories are observed, following, for example, the approach in [6].

Our approach follows on our previous work [7], using critical points for efficient and effective trajectory identification, alleviating the need for time alignment. Our approach assumes that each segment in the observed trajectory is either the initial segment of a movement in its library or the continuation of the previous movement. Then, given a partial trajectory, our method is able to predict the most likely next target—i.e., the end-point of the movement currently being executed, if the latter is executed to the end. By using an effective search tree, our approach is able to run at execution time and provide an efficient way for action recognition and prediction.

References

1. Black, M., Jepson, A.: A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions. In: Proc. 10th Eur. Conf. Computer Vision. pp. 909–924 (1998)
2. Black, M., Jepson, A.: Recognizing temporal trajectories using the condensation algorithm. In: Proc. 3rd IEEE Int. Conf. Automatic Face and Gesture Recognition. pp. 16–21 (1998)
3. Dermy, O., Paraschos, A., Ewerton, M., Peters, J., Charpillet, F., Ivaldi, S.: Prediction of intention during interaction with iCub with probabilistic movement primitives. Frontiers in Robotics and AI **4**, 45 (2017)
4. Ijspeert, A., Nakanishi, J., Hoffmann, H., Pastor, P., Schaal, S.: Dynamical movement primitives Learning attractor models for motor behaviors. Neural Computation **25**(2), 328–373 (2013)

5. Kelso, J.: Dynamic Patterns: The Self-Organization of Brain and Behavior. MIT Press (1995)
6. Kordia, A., Melo, F.: An end-to-end approach for learning and generating complex robot motions from demonstration. In: Proc. 16th IEEE Int. Conf. Control, Automation, Robotics and Vision. pp. 1008–1014 (2020)
7. Kordia, A., Melo, F.: Movement recognition and prediction using DMPs. In: Proc. 2021 IEEE Int. Conf. Robotics and Automation (2021)
8. Kulvicius, T., Ning, K., Tamosiunaite, M., Worgötter, F.: Joining movement sequences: Modified dynamic movement primitives for robotics applications exemplified on handwriting. IEEE Trans. Robotics **28**(1), 145–157 (2012)
9. Lee, D., Ott, C., Nakamura, Y.: Mimetic communication model with compliant physical contact in human-humanoid interaction. Int. J. Robotics Research **29**(13), 1684–1704 (2010)
10. Loshchilov, I.: A computationally efficient limited memory CMA-ES for large scale optimization. In: Proc. 2014 Annual Conf. Genetic and Evolutionary Computation. pp. 397–404 (2014)
11. Ma, S., Zhang, J., Ikizler-Cinbis, N., Sclaroff, S.: Action recognition and localization by hierarchical space-time segments. In: Proc. IEEE Int. Conf. Computer Vision. pp. 2744–2751 (2013)
12. Maeda, G., Neumann, G., Ewerton, M., Lioutikov, R., Kroemer, O., Peters, J.: Probabilistic movement primitives for coordination of multiple human-robot collaborative tasks. Autonomous Robots **41**(3), 593–612 (2017)
13. Meier, F., Theodorou, E., Schaal, S.: Movement segmentation and recognition for imitation learning. In: Proc. 15th Int. Conf. Artificial Intelligence and Statistics. pp. 761–769 (2012)
14. Mulling, K., Kober, J., Peters, J.: Learning table tennis with a mixture of motor primitives. In: Proc. 2010 IEEE-RAS Int. Conf. Humanoid Robots. pp. 411–416 (2010)
15. Nemec, B., Tamosiunaite, M., Woergoetter, F., Ude, A.: Task adaptation through exploration and action sequencing. In: Proc. 9th IEEE-RAS Int. Conf. Humanoid Robots. pp. 610–616 (2009)
16. Oliver, N., Rosario, B., Pentland, A.: A Bayesian computer vision system for modeling human interactions. IEEE Trans. Pattern Analysis and Machine Intelligence **22**(8), 831–843 (2000)
17. Stulp, F., Sigaud, O.: Path integral policy improvement with covariance matrix adaptation. In: Proc. 29th Int. Conf. Machine Learning. pp. 1547–1554 (2012)
18. Stulp, F., Sigaud, O.: Robot skill learning: From reinforcement learning to evolution strategies. PALADYN J. Behavioral Robotics **4**(1), 49–61 (2013)
19. Tanaka, Y., Kinugawa, J., Sugahara, Y., Kosuge, K.: Motion planning with worker's trajectory prediction for assembly task partner robot. In: Proc. 2012 IEEE/RSJ Int. Conf. Intelligent Robots and Systems. pp. 1525–1532 (2012)
20. Yoon, H., Soh, J., Bae, Y., Yang, H.: Hand gesture recognition using combined features of location, angle and velocity. Pattern Recognition **34**(7), 1491–1501 (2001)